Stanford Machine Learning — Support Vector Machine

1. Large Margin Classification.

Optimization Objective.

- logistic regression $h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$.

  - if $y=1$, we want $h_\theta(x) \approx 1 \Rightarrow \theta^T x \gg 0$

    $y=0$, $h_\theta(x) \approx 0 \Rightarrow \theta^T x \ll 0$.

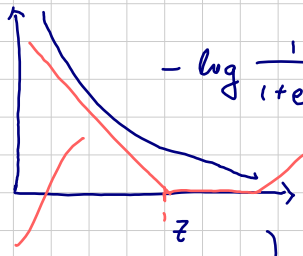
- cost of example: $-[y \log h_\theta(x) + (1-y) \log (1-h_\theta(x))]$

$$= -y \log \frac{1}{1+e^{-\theta^T x}} - (1-y) \log \left(1 - \frac{1}{1+e^{-\theta^T x}}\right)$$

① ②

① If $y=1$ (want $\theta^T x \gg 0$)

$z = \theta^T x$.

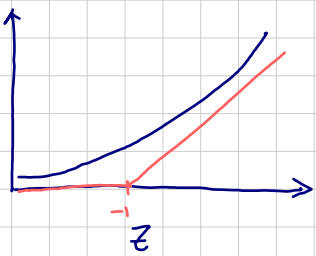


support vector machine.

straight line portion

large z, small cost contribution.

② If $y=0$, we want ($z = \theta^T x \ll 0$)



Support Vector Machine

· Logistic Regression.

$$\min_\theta \frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)}\underbrace{(-\log h_\theta(x^{(i)}))}_{\text{cost}_1(\theta^T x^{(i)})} + (1-y^{(i)})\cdot\underbrace{(-\log(1-h_\theta(x^{(i)})))}_{\text{cost}_0(\theta^T x^{(i)})}\right] + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2$$

· Support Vector Machine.

$$\min_\theta \frac{1}{m}\cdot\sum_{i=1}^{m} y^{(i)}\cdot\text{cost}_1(\theta^T x^{(i)}) + (1-y^{(i)})\cdot\text{cost}_0(\theta^T x^{(i)}) + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2$$

· convention : (1) remove $(\frac{1}{m})$ term → does not change $\underline{\theta}$ min

(2) log. reg format $A + \lambda B$.

S.V.M format $CA + B$ (diff. way of parametrization of tradeoffs)

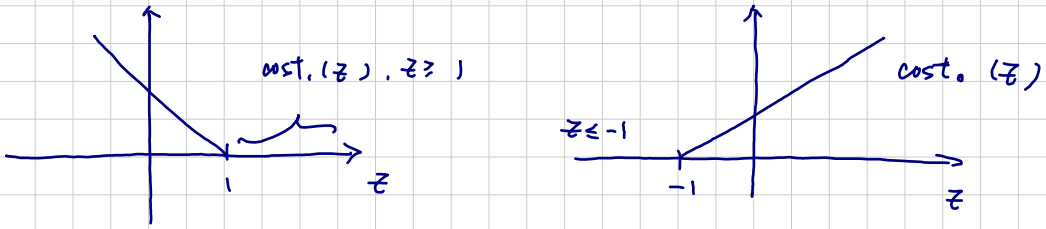
⇒ Support Vector Machine

$$\boxed{\min_\theta C\sum_{i=1}^{m}\left[y^{(i)}\cdot\text{cost}_1(\theta^T x^{(i)}) + (1-y^{(i)})\cdot\text{cost}_0(\theta^T x^{(i)})\right] + \frac{1}{2}\sum_{j=1}^{n}\theta_j^2}$$

· Hypothesis $h_\theta(x) = \begin{cases} 1, & \text{if } \theta^T x \geq 0 \\ 0, & \text{otherwise} \end{cases}$ ↪ form of hypothesis.

Large Margin Intuition

- Recall :



- If $y=1$, we want $\theta^T \cdot x \geq 1$ (not just ≥ 0)

  $y=0$, we want $\theta^T \cdot x \leq -1$ (not just < 0)

  ↓

  for logistic regression

- Here, we shift the threshold to be more conservative.

- SVM Decision Boundary.

  (1) setting $C = 10,000$.

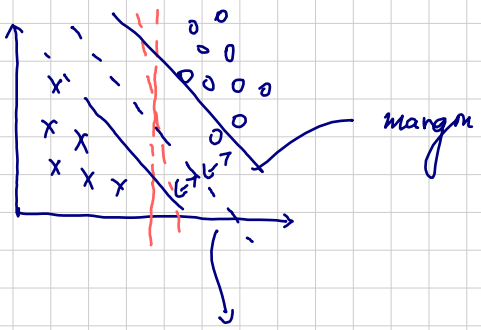
$$\min_\theta \; C \sum_{i=1}^{m} \left[y^{(i)} \text{cost}_1(\theta^T x^{(i)}) + (1-y^{(i)}) \text{cost}_0(\theta^T \cdot x^{(i)}) \right] + \frac{1}{2} \sum_{j=1}^{n} \theta_j^2$$

Whenever $y^{(i)}=1$, we need $\theta^T x^{(i)} \geq 1$

$y^{(i)}=0$, $\theta^T x^{(i)} \leq -1$

$$\Rightarrow \min \; C \cdot 0 + \frac{1}{2} \sum_{j=1}^{n} \theta_j^2 \quad \text{s.t.} \quad \theta^T x^{(i)} \geq 1 \text{ if } y^{(i)}=1, \quad \theta^T \cdot x^{(i)} \leq -1 \text{ if } y^{(i)}=0$$

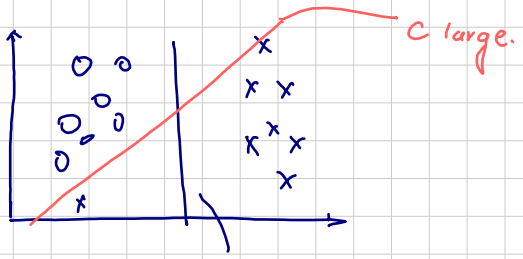
SVM Decision Boundary: Linearly Separable Case.



SVM = Large Margin classifier.

larger min. distance to data points. (large margin)

Large Margin classifier in Presence of Outliers



C not too large , C similar to $\frac{1}{\lambda}$ if λ small, then we have overfitting due to outliers.

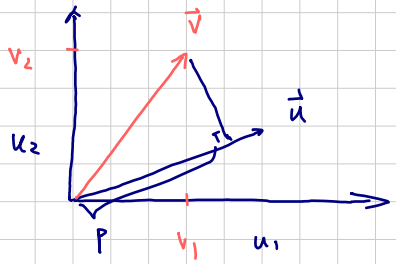
Mathematics Behind Large Margin Classification

1 Review.

① Vector inner product.

$$\vec{u} = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \quad \vec{v} = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}$$

inner product = $u^T \cdot v$.



$\|\vec{u}\|$ = length of $\vec{u}$ = $\sqrt{u_1^2 + u_2^2} \in \mathbb{R}$.

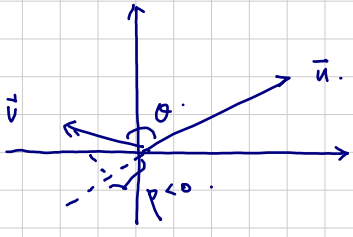
p = length of projection of $\vec{v}$ onto $\vec{u}$. (signed).

$u^T v = p \cdot \|\vec{u}\|$. (Can be shown)

so. $\begin{pmatrix} u_1 \\ u_2 \end{pmatrix}^T \cdot \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = p \cdot \|\vec{u}\| \quad \in \mathbb{R}^2$

$\Rightarrow$ $\boxed{u_1 v_1 + u_2 v_2 = p \cdot \|\vec{u}\|}$ Theorem.

· Sign of p.



$p<0$ if $\theta > 90°$.

2. SVM Decision Boundary

$$\min_{\theta} \ \frac{1}{2} \sum_{j=1}^{n} \theta_j^2$$

s.t. $\theta^T \cdot x^{(i)} \geq 1$ if $y^{(i)} = 1$

$\theta^T \cdot x^{(i)} \leq -1$ $y^{(i)} = 0$

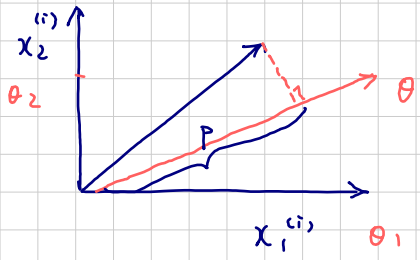
simplification $\theta_0 = 0$, $n = 2$ $\Rightarrow$ $\min_{\theta} \frac{1}{2} \sum_{j=1}^{n} \theta_j^2 = \frac{1}{2}(\theta_1^2 + \theta_2^2)$

$= \frac{1}{2}\left(\sqrt{\theta_1^2 + \theta_2^2}\right)^2$

$= \frac{1}{2} \|\theta\|^2$, where $\theta = \begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix}$

$\theta^T x^{(i)} = ?$

u^T v



$\theta^T x^{(i)} = p^{(i)} \cdot \|\theta\|$

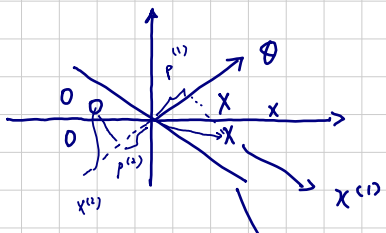
$= \theta_1 x_1^{(i)} + \theta_2 x_2^{(i)}$

$\Rightarrow$ $\min_{\theta} \frac{1}{2} \sum_{j=1}^{n} \theta_j^2 = \frac{1}{2} \|\theta\|^2$ s.t. $p^{(i)} \cdot \|\theta\| \geq 1$, if $y^{(i)} = 1$

$p^{(i)} \cdot \|\theta\| \leq -1$, if $y^{(i)} = 1$

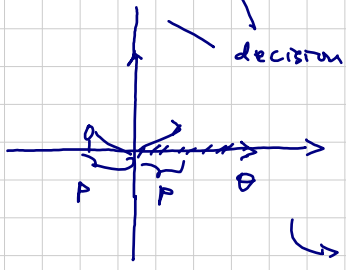
Where $p^{(i)} = \text{proj}_{\theta} x^{(i)}$

simplification.



$p^{(i)}$ small, but $p^{(i)} \cdot \|\theta\| \geq 1$

$\Rightarrow$ need large θ $\longleftrightarrow$ min. $\frac{1}{2}\|\theta\|$ conflict.



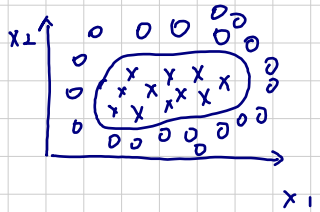
Now $p^{(i)}$ bigger, we can have a smaller θ $\Rightarrow$ min. $\frac{1}{2}\|\theta\|^2$ ✓

$\hookrightarrow$ SVM choose this decision boundary

Simplification. $\theta_0 = 0$ $\Rightarrow$ decision passes through boundary.

2. Kernels.

· Non-linear decision boundary



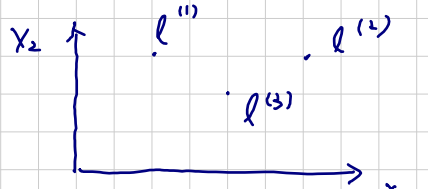
$h_\theta(x) = \begin{cases} 1 & , \text{ if } \theta_0 + \theta_1 x_1 + \cdots \geq 0 \\ 0 & , \text{ otherwise.} \end{cases}$

⇒ can be written as $\theta_0 + \theta_1 f_1 + \theta_2 f_2 + \theta_3 f_3 + \cdots$

$f_1 = x_1$, $f_2 = x_2$, $f_3 = x_1 x_2$, $f_4 = x_1^2$, $f_5 = x_2^2 \ldots$

↳ Is there a better choice of features $f_1, f_2, \ldots$? (high order polynomials are expensive)

· Kernel.



Given x, compute new feature depending on proximity to landmarks $l^{(1)}$, $l^{(2)}$, $l^{(3)}$.

$f_1 = \text{similarity}(x, l^{(1)})$ ← square of euclidean distance.

$= \exp\left(- \frac{\| x - l^{(1)} \|^2}{2\sigma^2} \right)$

⋮

kernel. ◁— $f_i = \text{similarity}(x, l^{(i)})$

$k(x, l^{(i)})$ $= \exp\left(- \frac{\| x - l^{(i)} \|^2}{2\sigma^2} \right)$

↳ A specific type of kernel (gaussian).

· Kernels and Similarity.

· note: $\| x - l^{(i)} \|^2$ can be written as $\sum_{j=1}^{n} (x_j - l_j^{(i)})^2$

(1) If $x \approx l^{(1)}$, then $\| x - l^{(1)} \|^2 \approx 0$.

$\Rightarrow \boxed{f_1 \approx \exp\left(- \frac{0}{2\sigma^2} \right) = 1}$

(2) If x far. $l^{(1)}$, then $\| x - l^{(1)} \|^2$ large

$\boxed{f_1 \approx 0}$

· Each landmark defines a new feature. $l^{(i)} \leftrightarrow f_i$

Ex.

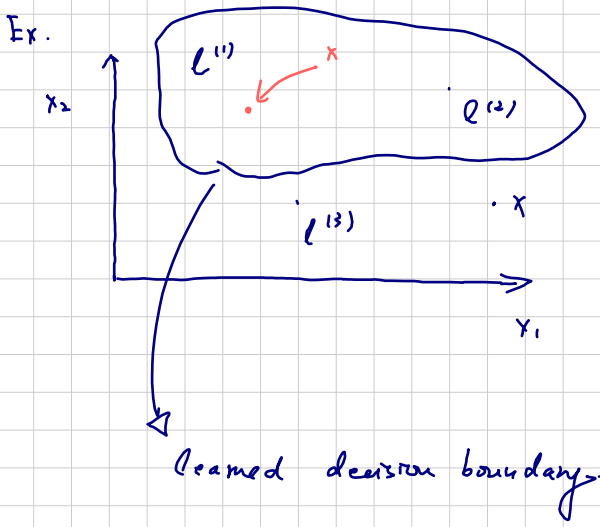
· $l^{(1)} = \begin{pmatrix} 3 \\ 5 \end{pmatrix}$ $\quad f_1 = \exp\left(- \frac{\| x - l^{(1)} \|^2}{2\sigma^2} \right)$, $\sigma^2 = 1$.

(variance).

· as σ^2 incr, f falls slower.

decr. faster

Ex.



learned decision boundary

Predict "1" when

$$\theta_0 + \theta_1 f_1 + \theta_2 f_2 + \theta_3 f_3 \geq 0 \quad (*)$$

① $f_1, f_2, f_3 \longrightarrow x$.

$\theta_0 = -0.5$, $\theta_1 = 1$, $\theta_2 = 1$, $\theta_3 = 0$.

$(*) \Rightarrow$ $f_1 \approx 1$ (x close to $l^{(1)}$)

$f_2 \approx 0$, $f_3 \approx 0$

$\Rightarrow \theta_0 + \theta_1 \cdot 1 + \theta_2 \cdot 0 + \theta_3 \cdot 0 = -0.5 + 1$

$= 0.5 \geq 0 \Rightarrow$ predict "1"

$(*) \Rightarrow$ $f_1, f_2, f_3 = 0$

$\theta_0 + \cdots = -0.5 < 0 \Rightarrow$ predict "0"

$\Rightarrow$ for points close to $l^{(1)}, l^{(2)} \rightarrow$ predict 1.

far. 0

· Choosing landmark.

- Where do we get $l^{(1)}, l^{(2)}, l^{(3)}$?

predict $y=1$ if $\theta_0 + \theta_1 f_1 + \theta_2 f_2 + \dots \geq 0$.

- features are measuring on how close to training examples, given a x.

- Given. $(x^{(1)}, y^{(1)})$, $(x^{(2)}, y^{(2)})$, $(x^{(3)}, y^{(3)})$ ... $(x^{(m)}, y^{(m)})$

$\Rightarrow$ choose $l^{(1)} = x^{(1)}$, $l^{(2)} = x^{(2)}$, ..., $l^{(m)} = x^{(m)}$.

- given example x,

$f_1 = \text{similarity}(x, l^{(1)})$ → $x^{(1)}$.

$f_2 = \text{similarity}(x, l^{(2)})$ → $x^{(2)}$

$\vdots$

$\Rightarrow f = \begin{bmatrix} f_0 \\ f_1 \\ \vdots \\ f_m \end{bmatrix}$ $f_0 = 1$.

→ each element rep. how close the vector x is to each landmark. $l^{(i)}$.

· For training example $(x^{(i)}, y^{(i)})$

$x^{(i)} \longrightarrow$ $f_1^{(i)} = \text{sim}(x^{(i)}, l^{(1)})$

$f_2^{(i)} = \text{sim}(x^{(i)}, l^{(2)})$

$\vdots$

$f_m^{(i)} = \text{sim}(x^{(i)}, l^{(m)})$

$\longleftarrow$ $f_i^{(i)} = \text{sim}(x^{(i)}, l^{(i)})$ ← $x^{(i)}$

$= \text{sim}(x^{(i)}, x^{(i)})$

$= 1$.

$x^{(i)} \in \mathbb{R}^{n+1}$ $(0, \dots n)$ $\Rightarrow$ $f^{(i)} = \begin{pmatrix} f_0^{(i)} \\ f_1^{(i)} \\ \vdots \\ f_m^{(i)} \end{pmatrix}$ $\boxed{f_0^{(i)} = 1}$

SVM with Kernels

① Hypothesis: given x, compute features $f \in \mathbb{R}^{n+1}$.

predict "$y = 1$" if $\theta^T f \geq 0$ $\Rightarrow$ $\sum_0^m \theta_i f_i \geq 0$

② Training: $\min_\theta C \sum_{i=1}^{m} y^{(i)} \cdot \text{cost}_1(\theta^T \cdot f^{(i)}) +$

$(1 - y^{(i)}) \cdot \text{cost}_0(\theta^T \cdot f^{(i)}) + \frac{1}{2} \sum_{j=1}^{m} \theta_j^2$ ($n = m$!)

③ Implementation note.

- $\sum_{j=1}^{m} \theta_j^2 = \theta^T \cdot \theta$, $\theta = \begin{pmatrix} \theta_1 \\ \vdots \\ \theta_m \end{pmatrix}$ → ignoring θ_0.
- often, one can also scale the θ. i.e. $\theta^T \cdot \underline{M} \theta$.
- $M \rightarrow$ ↑ computation efficiency.

SVM Parameters

$C \left(= \frac{1}{\lambda}\right)$ < large C: low bias, high variance (small λ)
small C: high bias, low variance. (large λ)

σ^2 Large σ^2: features f_i vary more smoothly.

High bias, lower variance (changes too slow)


Gaussian kernel

$\exp\left(-\frac{\|x - \ell^{(i)}\|^2}{2\sigma^2}\right)$

small σ^2 . features f_i vary less smoothly.

lower bias, high variance

(changes abruptly).

3. SVM in Practise. : Using a SVM.

. Use SVM software package (eg. liblinear, libsvm)

to solve for SVM.

. Specification

(1) Choice of parameter C.

(2) Choice of kernal. (similarity function)

a. No kernel. ("linear kernel")

. Predict $y=1$ if $\theta^T x > 0$

↳ good for large n (# of features) $x \in \mathbb{R}^{n+1}$.

small m. (small data set)

b. Guassian Kernel.

① $f_i = \exp\left(-\frac{\|x - l^{(i)}\|^2}{2\sigma^2}\right)$, where $l^{(i)} = x^{(i)}$.

Need to choose σ^2

↳ good for small n. ($x \in \mathbb{R}^{n+1}$)

and/or large m.

② Kernel (similarity function)

function $f = \text{kernel}(x1, x2)$

$f = \exp\left(-\frac{\|x1 - x2\|^2}{2\sigma^2}\right)$

return.

* Note: perform feature scaling before using the Guassian kernel.

C. Other choices of kernel.

. not all similarity functions similarity (x, ℓ) make valid kernels.

[Need to classify technical condition called "Mercer's theorem" to make sure SVM packages' optimizations run correctly and not diverge]

. Available options.

(1) polynomial kernel $k(x, \ell) = (x^T \ell)^2$,

$(x^T \ell)^3$, $(x^T \ell + 1)^3$.

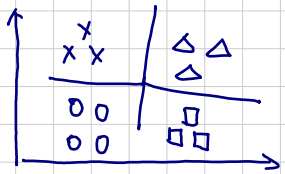
$(x^T \ell + 5)^4$

two parameters c. d

s.t. $(x^T \ell + c)^d$.

(2) Esoteric: string kernels (text), chi-square kernel, histogram kernel, ...

. Multiclass classification.



$y \in \{1, 2, 3 \dots K\}$.

Δ pick class i with largest $(\theta^{(i)})^T x$.

. Logistic Regression vs. SVMs.

n: # features. m = # examples

(1) $n \gg m$. eg. $n = 10000$, $m = 10 \dots 1000$.

- logistic regression
- SVM w/o kernel. (linear kernel)
- not enough data.

(2) n small, m intermediate $n = 1 \dots 1000$.
$m = 10, \dots 10,000$

- SVM with Guassian Kernel.

(3) n small, m large. $n = 1 - 1000$, $m = 50,000+ \dots$

- Create / add more features

↓

use → logistic regression
SVM without a kernel.

→ similar algorithm (similar output, slight difference in efficiency)

- Neural network: likely to work with any, may be slower to train.